DAA ASSIGNMENT-2

By:

Gitika Yadav - IIT2019219

Divyatez Singh - IIT2019220

Divyansh Rai - IIT2019221

Question:

Problem Statement:

Use divide and conquer to compute 1+2+3+4+.... upto n numbers.

Aspect of divide and conquer:

DIVIDE: This involves the dividing the problem into subproblems.

CONQUER: Sub problem by calling recursively until sub problem solved.

COMBINE: The result of several sub problems are combined to generate the final output according to the requirement of question.

Approach:

1. We store values from 1 to n in an array

2. In sumArray function we check for base cases like if size of array is 0, then sum is 0.

3. Else if size of array is 1, then sum is value of the only element.

4. Then we apply divide and conquor by dividing array in two halves and recursively calling the function sumArray.

5. Finally, we return the sum of the two halves.

```cpp
#include<bits/stdc++.h>
using namespace std;

int sumArray(int anArray[], int size)
{
    //base case
    if (size == 0)
    {
        return 0;
    }
    else if (size == 1)
    {
        return anArray[0];
    }

    //divide and conquer
    int mid = size / 2;
    int rsize = size - mid;
    int lsum = sumArray(anArray, mid);
    int rsum = sumArray(anArray + mid, rsize);
    return lsum + rsum;
}

int main(){
    int n;
    cout<<"Enter a number : ";
    cin>>n;
    int arr[n];
    for(int i=1;i<=n;i++)
    {
        arr[i-1]=i;
    }
    cout<<"Sum will be : "<<sumArray(arr,n)<<endl;
}
```

TEST CASES:

Test Case-1

```
Input:          10

Output:         55

*************************************************************
```

Test Case-2

```
Input:          25

Output:         325

*************************************************************************
```

Time and space complexity:

TIME COMPLEXITY:

T(n)=2T(n/2)+O(1)

Using above relation , we get for (T/2) , (T/8) etc. as:

T(n/2)=2T(n/4)+O(1)

T(n/4)=2T(n/8)+O(1)

T(n/8)=2T(n/16)+O(1) ans so on….

On combining we get the overall time complexity as: T(n)=O(n)

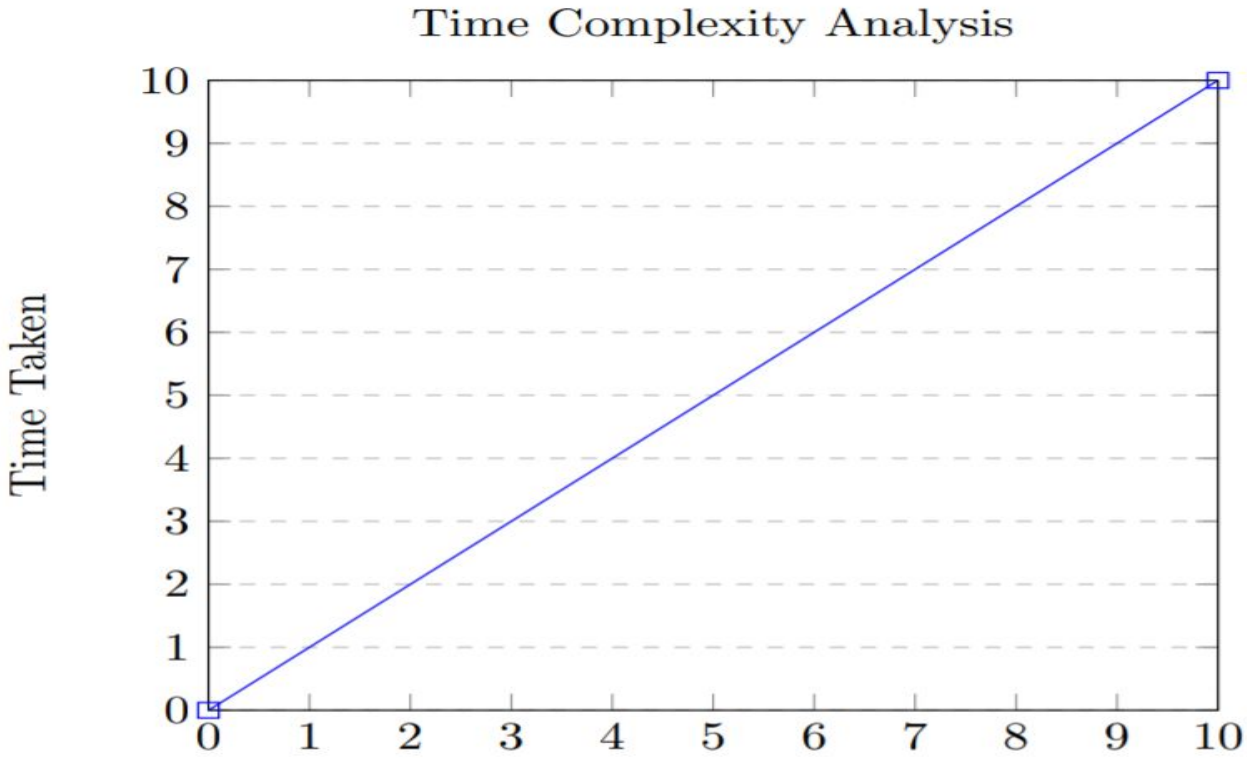
****

SPACE COMPLEXITY:

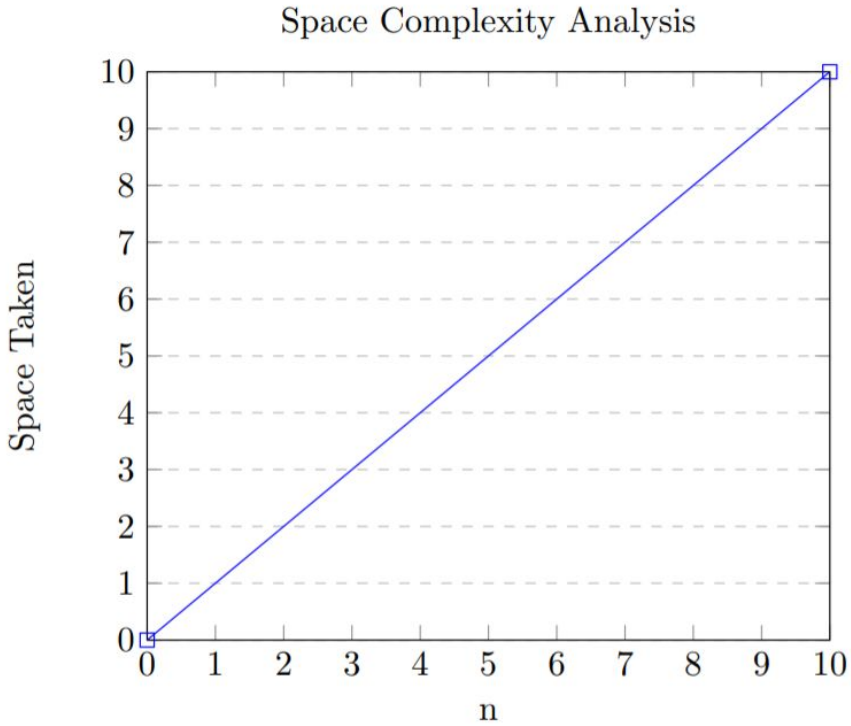
For all the cases : O(n)

****

GRAPH ANALYSIS 1:



GRAPH ANALYSIS 2:



THANK YOU